

Coding Agents as Design Searchers: An Autonomous TAGE Tuning Campaign for CBP-NG

Matthew Pallan
University of Maryland
College Park, MD, USA
mtp@umd.edu

Abstract—Autonomous, general-purpose coding agents produced this CBP-NG branch-predictor submission, and this paper argues that the method of production, not the predictor, is the transferable contribution. A primary agent (Claude Code) drove the simulator, edited the predictor source, ran the screening pipeline, selected the next candidate, and drafted this paper; a second agent (OpenAI Codex) generated the figures and ran an independent, adversarial architecture search. Both operated under a single standing instruction, to record every experiment in a dated journal and a machine-readable ledger, which served as the agents’ shared memory across restarts. Over roughly three weeks the agents evaluated more than 2,300 configurations across 237 sessions, including one stretch of about thirty hours of wall-clock across three back-to-back runs, promoting a candidate only after confirmation at full simulation length. The resulting pure-TAGE design reaches VFS 0.9345, within 1.3% of the 0.946 ceiling. We treat the predictor as a case study: its three findings (that throughput dominates accuracy under VFS, that the statistical corrector is infeasible under HARCOT, and that training-dynamics changes are mirages at medium evaluation length) show that the autonomous loop did real work, and we do not claim them as novel mechanisms. We close by examining the method’s limitations, which are properties of present-day models more than of the method itself, and the gains a purpose-built research harness could add.

Index Terms—branch prediction, autonomous agents, design-space search, large language models, hardware tuning, CBP-NG.

I. Introduction

The usual object of a championship branch-prediction paper is a predictor. This paper is chiefly about a process: it asks how far a pair of general-purpose AI coding agents, given an objective and a small set of constraints, can carry the design of a competitive predictor on their own. The predictor they produced is competitive but incremental; the method that produced it is the part worth reporting.

CBP-NG evaluates conditional branch predictors using VFS (voltage-frequency-scaled speedup) [1], which combines prediction throughput (IPC_{cbp}), misprediction penalty (CPI_{cbp}), and dynamic energy (EPI_{cbp}):

$$VFS = \frac{IPC_{cbp}}{IPC^*} \cdot \frac{CPI^*}{CPI_{cbp}} \cdot \left(\frac{EPI^*}{EPI_{cbp}} \right)^{1/\gamma} \quad (1)$$

where $\gamma=3.2$ and $*$ denotes reference values ($IPC^*=8$, $CPI^*=0.0315$, $EPI^*=1000$). Unlike prior championships,

which optimised accuracy alone under a storage budget, CBP-NG requires designs to be implemented in HARCOT [2], a C++ hardware-modelling library that accounts for gate timing, wire delays, RAM access conflicts, and chip floorplan area. Two features of this setting make it an unusually good fit for autonomous search. First, the objective is a single scalar that a simulator computes deterministically and cheaply, so an agent can score every candidate it proposes without human judgement in the loop. Second, the predictor is expressed as a template with some fifty-seven parameters, so a candidate is a short, mechanical edit. An agent can therefore propose, build, evaluate, and record a design in one uninterrupted cycle.

This paper makes one methodological contribution and uses one case study to support it. The contribution is an account of an autonomous, fully-logged, agent-driven design search: its orchestration (Section II), the self-correcting screening heuristic that emerged from its own records (Section III), and an honest appraisal of where it failed and what would improve it (Section V). The case study is the predictor the loop produced (Section IV), whose three findings we summarise here only to establish that the search did real work, and not to claim the mechanisms as novel; they are incremental refinements of established TAGE [3] techniques.

- 1) Throughput dominates accuracy under VFS. A single block-size change (LOGLB 6→7, IPC 5.6→6.8) supplied roughly 65% of the total gain, while accuracy and energy together supplied the remainder, of which only some 4% was attributable to accuracy-specific mechanisms (Section IV).
- 2) The statistical corrector is infeasible under HARCOT. Seznec’s SC [4], [5] has won every CBP since 2014, yet under hardware-realistic modelling it cannot be fitted within the two-cycle second-level budget without either breaching the latency bound or sacrificing accuracy to aliasing (Section IV).
- 3) Medium-length screening misleads for training-dynamics changes. Of 31 changes evaluated at both horizons, 13 of 16 training-dynamics changes collapsed at full length while 14 of 15 structural

changes transferred (87% classification accuracy), a regularity the agents discovered from their own log and then exploited (Section III).

Guided by these findings, the pure TAGE design reaches VFS 0.9345, within 1.3% of the 0.946 theoretical ceiling.

II. The Orchestration Method

A. Roles and division of labour

Two general-purpose coding agents carried out the search, with distinct roles. The primary agent (Claude Code) held the main loop: it proposed a mechanism, expressed it as a change to the predictor’s template parameters or source, compiled it, ran the simulations, recorded the outcome, and chose the next candidate. The same agent wrote the supporting documentation and successive drafts of this paper. A second agent (OpenAI Codex) was used for two narrower purposes: generating and refining the figures, and conducting an independent architecture search whose role was deliberately adversarial, attacking the primary’s apparent gains and looking for structural ideas the primary had not found (Section II-D).

The human supplied only three things, fixed early and rarely revisited: the objective (maximise VFS on the public traces), the hard physical constraints (Section II-C), and a single standing rule about record-keeping. Within those bounds the agents ran without step-by-step direction; they committed their own work, and human feedback stayed occasional and cosmetic, such as corrections to figure layout. All experiments drew on the 168 public CBP-NG training traces.

B. The journal as shared memory

The one indispensable rule was about memory. After every experiment or decision, the agent appended a dated entry to a journal (`predictor_progress.md`) and a row to a machine-readable ledger (`predictor_sweeps.csv`). This was not documentation for its own sake: because an agent’s context window cannot hold a three-week search, the journal was the search state. Each session reconstructed what had been tried, what had won, and what had lost by reading the journal, and each restart resumed from it. By the end the journal ran to some 12,700 lines across 237 numbered sessions, and the ledger held every configuration the agents had scored. The cost of this discipline was small; its payoff, a regularity invisible in any single experiment but plain in the aggregate, is the subject of Section III.

C. A cost-tiered promotion gate

Each session advanced a single hypothesis through a pipeline of increasing cost (Figure 1). A candidate was screened first at short length (10K warmup / 100K simulated instructions on a few traces), promoted on survival to medium length (100K / 500K on all 168 traces), and only then, if it remained promising and was structurally motivated, to the official length (1M / 40M on all 168 traces). A change entered the design only after

confirmation at official length. Of the more than 2,300 configurations the agents screened, only 31 reached full-length evaluation and 6 mechanisms survived into the final design (Figure 3). The hard constraints lived at the gate as well: second-level latency was held at two cycles, since a third cycle removes 4–5% of VFS at a stroke; tag width was held at no fewer than twelve bits; and the table-geometry parameters that breach the two-cycle bound were fixed at their limits.

D. Cross-model escalation and adversarial checking

The second model addressed two failure modes of a single agent in a loop. The first is the plateau: once the primary agent had exhausted parameter tuning at VFS 0.934, it could no longer find improvements, and continued search merely re-confirmed dead ends. At that point the primary wrote a self-contained briefing (the current design, the binding constraints, the catalogue of what had already failed, and a set of speculative directions) and handed it to the second agent to attempt an architectural breakthrough from a fresh vantage. The second failure mode is credulity: an agent that proposes and scores its own ideas tends to believe marginal results. The second agent was therefore also used adversarially, re-testing the primary’s near-zero “gains” (for instance, a reported +0.000009 VFS from stacking two changes) and repeatedly showing them to be noise. Neither mechanism broke the 0.934 plateau, which we take to be a genuine design-space boundary, not a search boundary (Section IV); both, however, were essential to keeping the search honest.

E. Autonomy in practice

The search was not a sequence of short supervised prompts but a handful of long, self-directed runs, restarted by hand when they ended (Figure 2). Within a run the agent committed a result every few minutes to roughly an hour (median twelve minutes); the longer pauses coincide with full-length validations, which take hours, so the loop was in effect blocking on its own most expensive gate. Across the campaign the agents ran about ten substantial unattended sessions. The longest single uninterrupted run lasted some eleven hours; the most intensive stretch covered sessions 49–235 across three back-to-back runs spanning roughly thirty hours of wall-clock on 21–22 March.

In total the agents logged more than 2,300 configurations, the great majority of which did not survive. Every post-TAGE corrector that was tried breached the timing budget; several alternative architectures regressed, one doubling the misprediction rate; and most candidates that gained at medium length lost that gain at full length. Aggregated in the journal, these failures exposed a regularity no single run could show.

III. An Emergent Screening Heuristic

The study’s central methodological result was not designed; the agents read it off the accumulated jour-

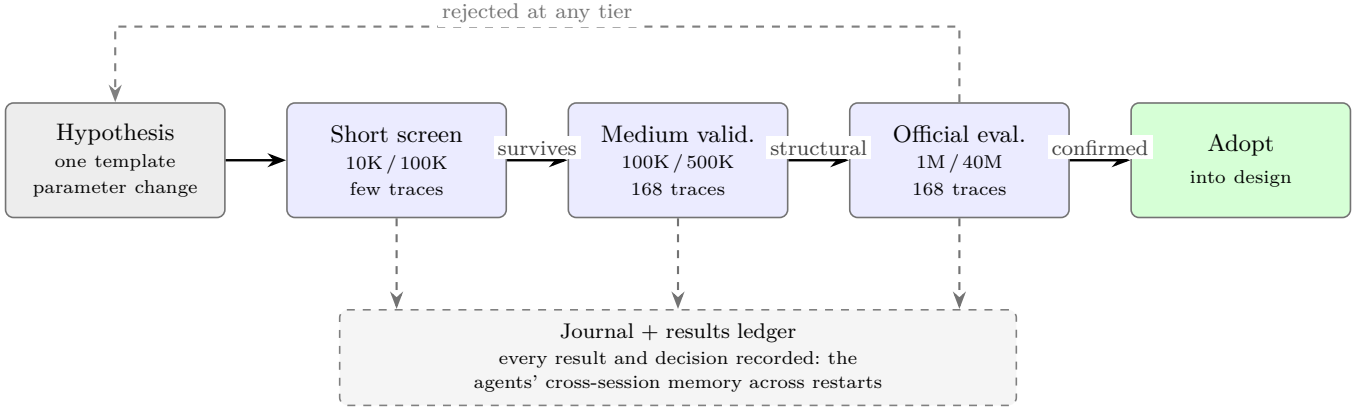


Fig. 1. The per-session search loop. A hypothesis, expressed as one template-parameter change, is promoted left to right through three tiers of increasing cost and adopted only after confirmation at full length; only structurally motivated candidates reach the official tier (Section III). Every outcome, adopted or rejected, is written to the journal and results ledger, which together form the agents’ cross-session memory.

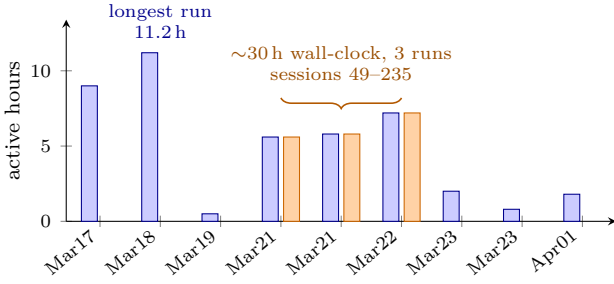


Fig. 2. Active hours per autonomous run, reconstructed from commit timestamps. Bar height is committing time within a run; gaps between runs are human restarts. The longest uninterrupted run lasted ~ 11 h; the orange bars are the 21–22 March push (sessions 49–235, ~ 30 h of wall-clock across three runs).

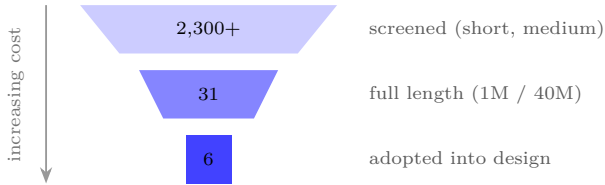


Fig. 3. Search economics under the promotion gate. Of more than 2,300 configurations screened, 31 earned full-length evaluation and 6 mechanisms were adopted into the final design.

nal. Changes to training dynamics tend not to transfer from medium to full-length evaluation, whereas structural changes transfer reliably (Figure 4). Each configuration was classified in advance as either structural (altering information content: hashing, indexing, table geometry, path context) or training-dynamics (altering convergence behaviour: update policy, counter width, initialisation). Of 31 configurations evaluated at both horizons, 14 of 15 structural changes transferred with a positive or amplified delta, while 13 of 16 training-dynamics changes collapsed; 27 of 31 were classified correctly (87%).

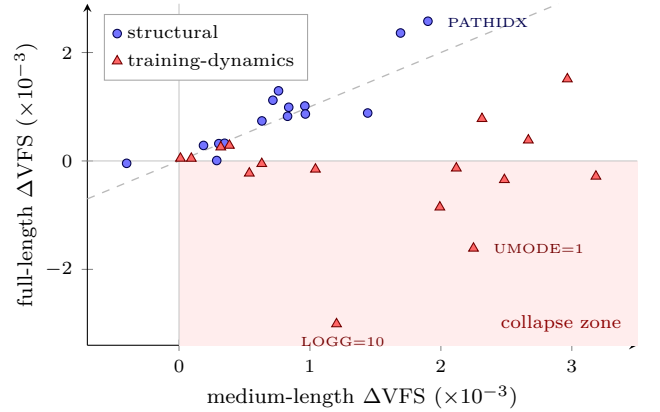


Fig. 4. Medium versus full-length VFS delta for 31 configurations. Structural changes (blue circles) lie on or above the $y = x$ line, transferring reliably and often with amplification; training-dynamics changes (red triangles) fall into the collapse zone, where a medium-length gain turns into a full-length loss.

The likely mechanism is that at 500K instructions the tables have not yet converged, so a change that merely accelerates convergence is indistinguishable from one that improves steady-state accuracy; by 40M instructions convergence has saturated and only the latter survives. Once the agents recognised this pattern in their own records, they adopted it as a screening rule, promoting only structurally motivated candidates to full-length evaluation. The rule skipped 16 of the 31 runs and saved roughly 80% of the full-length compute while missing only a single true positive. The search, in other words, used its own logged history to make itself cheaper. As Section V notes, this is also a hand-rolled instance of a problem with an established literature.

IV. Case Study: the Predictor the Loop Produced

We report the predictor as evidence that the autonomous search reached real conclusions. The final design

TABLE I
Final predictor metrics (168 traces, 1M warmup, 40M sim).

Metric	Value	Description
VFS	0.9345	Voltage-frequency-scaled speedup
IPC _{cbp}	6.844	Prediction throughput
CPI _{cbp}	0.0517	Misprediction penalty
EPI _{cbp}	1244 fJ	Dynamic energy per instruction
MPI	0.517%	Misprediction rate
L ₁	1 cycle	First prediction latency
L ₂	2 cycles	Second prediction latency

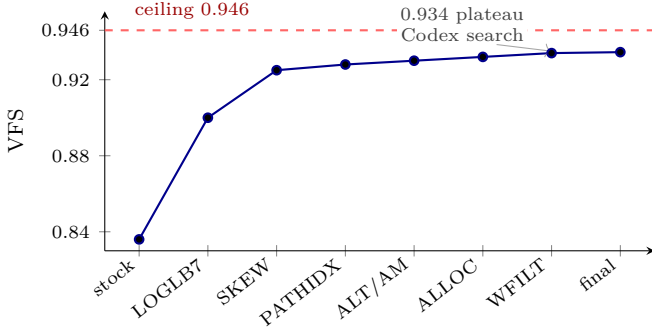


Fig. 5. VFS trajectory across the adopted milestones, from stock TAGE (0.836) to the final design (0.9345). The block-size change (LOGLB7) supplies most of the gain; the search then plateaus at 0.934, where the second agent was brought in. The dashed line is the $p2_lat=1$ ceiling.

is a TAGE [3] with 7 tagged tables (NUMG=7) of 2048 entries each (LOGG=11), a 16384-entry bimodal base (LOGB=14), a 128-byte (32-instruction) prediction block (LOGLB=7), and a fast gshare first level (P1). Its full-length metrics on all 168 traces appear in Table I; its development trajectory, from stock TAGE to the final design, appears in Figure 5.

Throughput dominates. VFS is far more sensitive to throughput than to accuracy or energy: a 10% improvement in IPC, MPI, and EPI is worth about +2.4%, +1.4%, and +0.2% VFS respectively, the energy term being damped by $\gamma=3.2$. The block-size change is the only lever in this design that moved IPC, raising it from 5.6 to 6.8; a term-by-term attribution places the split at roughly 65% throughput and 35% accuracy-and-energy, of which only about 4% is accuracy-specific. The conventional assumption that accuracy is paramount does not carry over to VFS-scored competition.

The SC is infeasible under HARCOM. The absence of a statistical corrector, present in every CBP winner since 2014, requires justification, and the search supplied three independent barriers, each on its own sufficient (Table II). The predict2 critical path already fills the two-cycle budget (approximately 600 ps at a 300 ps clock) with only about 18 ps of slack, recovered by compiling out dead RAM; an SC override MUX, the floorplan cost of its tables, or the aliasing of tables small enough to fit each break it. The result generalises: within this HARCOM-

TABLE II
Three independent barriers to SC in CBP-NG.

Barrier	Mechanism	Impact
Serial logic	Override MUX	+25–77 ps $\rightarrow L_2=3$
Floorplan area	4 rwrw decl.	+28–44 ps $\rightarrow L_2=3$
Table aliasing	32–128 entries	MPI $\times 2.4$

constrained space, no post-TAGE serial corrector (SC, perceptron override, or loop predictor) fits the two-cycle envelope.

The accuracy refinements that survived are modest: a path-register index decorrelation (PATHIDX, +0.0024 VFS, the largest single gain), a fix to a bimodal-hysteresis pollution effect (+0.0010), the 4-way SKEW hash (+0.0003), and energy-only write filtering. Their combined contribution is under +0.004 VFS, consistent with the throughput attribution above.

Distance to the ceiling. The theoretical VFS ceiling is 0.946, a $p2_lat=1$ counterfactual holding MPI and EPI at the design’s values while setting L_2 to one cycle; the present 0.9345 is within 1.3%, and the entire remaining gap is attributable to $L_2=2$, not $L_2=1$. The gap concentrates in 36 traces (21%) that are structurally hard: owing to energy scaling at $\gamma=3.2$, they cannot exceed VFS 0.95 even under perfect prediction. That every parameter sits at a physical boundary, and that the gap reduces to a single latency cycle, is itself a product of the exhaustive logged search.

V. Discussion

A. When this approach works

The campaign succeeded because the problem had a particular shape, and that shape predicts where the method transfers. Three properties were decisive: a cheap, deterministic scalar objective, so the agent could score its own proposals without human judgement; a candidate that was a small, mechanical edit to a template, so building one was fast and safe; and a trusted evaluator, so a confirmed number could be believed. Problems that share these properties, such as much hardware-parameter tuning, compiler-flag and kernel auto-tuning, and similar searches with a simulator in the loop, should suit autonomous agents well. Problems where success is a matter of taste, where evaluation is expensive or noisy, or where a candidate is a large and risky change will not.

B. Limitations

The record exposes several limitations. First, the agents were credulous about noise: lacking any notion of statistical significance across the 168 traces, they repeatedly logged sub-0.0001 “improvements” as gains, which adversarial re-testing then overturned. Scoring more than 2,300 configurations against one fixed trace set is, moreover, a form of adaptive data analysis, which inflates false discoveries unless the holdout is deliberately protected [18], [19].

Second, the search was local: it tuned parameters tirelessly and thoroughly, but neither agent invented a genuinely new architecture, and the plateau at VFS 0.934 was never broken by cleverness, only confirmed as a physical limit. Third, the search was coordinate-wise, advancing mostly one parameter at a time, which is ill-suited to a space with strong interactions among its fifty-seven parameters. Fourth, the agents fell repeatedly into the medium-length mirage before recognising the screening heuristic of Section III, wasting full-length compute on changes that could not transfer. Finally, the loop depended on external scaffolding, a hand-maintained journal for memory and periodic human restarts, instead of running as a closed system.

Several of these are properties of the particular 2026-era models used (Claude Code as primary, OpenAI Codex as secondary), not of agent-driven search as such. The need for an external journal followed from a finite context window; the restarts, from a bounded autonomous horizon. Both constraints are already loosening: models released within months of this work report markedly longer unattended operation and far better use of persistent external memory. Anthropic’s Claude Fable 5, for example, is reported to benefit about three times as much as the prior generation from file-based memory on a published benchmark [6]. The limitation most likely to persist is the second: the boundary between the exhaustive local search these agents perform tirelessly and the architectural invention they did not attempt. A caveat for reproducibility follows from the same observation: an agent-produced result is pinned to a model version, and those versions are neither stable nor guaranteed to remain available.

C. What a purpose-built harness would add

The search ran inside a generic coding agent in a chat loop, and improvised structures that established methods already supply. Its journal is a rudimentary evolutionary program database: LLM-driven program-search systems pair a generator with a deterministic evaluator that discards confabulations and hold their state in a scored population, run asynchronously and island-structured [7], [8], with sample-efficient variants reaching state-of-the-art results in roughly 150 evaluations [9]; their one precondition, a cheap and trusted score, is what CBP-NG supplies. Its three-tier promotion gate is an ad-hoc successive-halving scheme: Hyperband allocates fidelity as a bandit and reports more than an order-of-magnitude speed-up over full-fidelity Bayesian optimisation [10], ASHA promotes asynchronously and removes the straggler that stalls a synchronous gate [11], and BOHB adds a Bayesian model atop it [12], while learning-curve extrapolation predicts full-length performance from a partial run, the medium-length mirage by its proper name [13].

Its coordinate-wise tuning is a known antipattern, dominated by designed experiments and joint optimisation whenever parameters interact [15]; model-based optimi-

sation with a random-forest surrogate (SMAC) is built for exactly the categorical, conditional, high-dimensional space of a fifty-seven-parameter template [14]. And its lone agent should be a panel: an orchestrator fanning candidates out to parallel workers [16], an independent verifier bank in place of a single adversarial pass, and agentic tree-search of the kind already assembled into end-to-end research harnesses [17]. Significance testing across the 168 traces, with the holdout protected against adaptive reuse [18], [19], would reject sub-0.0001 “gains” automatically. The present study shows that a generic agent can run such a search at all; a dedicated harness should run it faster, more thoroughly, and with the statistical guarantees the chat loop never had.

VI. Conclusion

We set out to see how far autonomous coding agents could carry the design of a competitive branch predictor, given an objective, a few hard constraints, and a single discipline of exhaustive record-keeping. The answer is: most of the way. A primary agent and an adversarial second agent, sharing a journal as memory and a cost-tiered gate for promotion, evaluated more than 2,300 configurations across 237 sessions (including one roughly thirty-hour wall-clock stretch) and produced a pure-TAGE design at VFS 0.9345, within 1.3% of the 0.946 ceiling. Along the way the search read a self-correcting screening heuristic off its own log, saving most of its full-length compute. The predictor’s findings (throughput dominates, SC is infeasible, training-dynamics changes are mirages) are evidence that the loop did real work, not novel mechanisms. The method’s limitations are real, but many should recede as today’s models improve; the clear next step is to move this kind of search out of a generic chat loop and into a purpose-built research harness.

References

- [1] P. Michaud, “A scoring metric for the CBP-NG championship,” Nov. 2025.
- [2] P. Michaud, “HARCOM: a C++ library for hardware complexity estimation,” 2024.
- [3] A. Seznec, “A new case for the TAGE branch predictor,” in Proc. MICRO, 2011.
- [4] A. Seznec, “TAGE-SC-L branch predictors again,” in Proc. JILP Workshop (CBP-5), 2016.
- [5] A. Seznec, “TAGE-SC for CBP2025,” in Proc. CBP Workshop, 2025.
- [6] Anthropic, “Claude Fable 5 and Claude Mythos 5,” Jun. 2026.
- [7] B. Romera-Paredes et al., “Mathematical discoveries from program search with large language models,” *Nature*, vol. 625, 2024.
- [8] A. Novikov et al., “AlphaEvolve: a coding agent for scientific and algorithmic discovery,” arXiv:2506.13131, 2025.
- [9] Sakana AI, “ShinkaEvolve: evolving programs for open-ended scientific discovery,” arXiv:2509.19349, 2025.
- [10] L. Li et al., “Hyperband: a novel bandit-based approach to hyperparameter optimization,” *JMLR*, vol. 18, 2018.
- [11] L. Li et al., “A system for massively parallel hyperparameter tuning,” in Proc. MLSys, 2020.
- [12] S. Falkner, A. Klein, and F. Hutter, “BOHB: robust and efficient hyperparameter optimization at scale,” in Proc. ICML, 2018.
- [13] A. Klein et al., “Learning curve prediction with Bayesian neural networks,” in Proc. ICLR, 2017.

- [14] F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Sequential model-based optimization for general algorithm configuration,” in Proc. LION, 2011.
- [15] V. Czitrom, “One-factor-at-a-time versus designed experiments,” The American Statistician, vol. 53, no. 2, 1999.
- [16] Anthropic, “How we built our multi-agent research system,” 2025.
- [17] Sakana AI, “The AI Scientist-v2: workshop-level automated scientific discovery via agentic tree search,” 2025.
- [18] C. Dwork et al., “The reusable holdout: preserving validity in adaptive data analysis,” Science, vol. 349, 2015.
- [19] A. Blum and M. Hardt, “The ladder: a reliable leaderboard for machine learning competitions,” in Proc. ICML, 2015.

participants can check whether inactive structures inflate their own measured timing.

Appendix A

Cost Analysis

This appendix details the hardware cost of each predictor component as reported by the HARCOT model.

TABLE III
Storage breakdown of the final predictor design.

Component	Entries	Width	Bits
Bimodal base (LOGB=14)	16384	2	32768
TAGE tables $\times 7$ (LOGG=11)	7×2048	17	244K
Tags (TAGW=12)		12	172K
Hysteresis (3-bit)		3	43K
Useful bits (1-bit)		1	14K
Direction (1-bit)		1	14K
P1 gshare (LOGP1=13)	8192	2	16384
Global history	1	160	160
Path register	1	11	11
AM table	32	1	32
Meta counter	1	5	5
Total			293K

TABLE IV
Critical path timing breakdown.

Path segment	Latency
P1 (gshare lookup)	300 ps (1 cycle)
P2 total	600 ps (2 cycles)
Address + fold XOR	120 ps
Tag RAM read	280 ps
Tag compare + priority	130 ps
Prediction extract	70 ps
Timing headroom (after dead elim.)	18 ps

Energy. EPI = 1244 fJ: TAGE reads (60%), wiring (30%), update writes (10%). Accuracy (MPI) improvements are $5\times$ more impactful for VFS than wiring reduction. Write filtering saved 42 fJ via deferred u-bit reads and 5 fJ via saturated hysteresis skip.

Dead tables. Of the 722 RAM instances, 520 (72%) were dead: struct members never accessed (128 SC tables, loop arrays, and filter tables), left as scaffolding by abandoned mechanisms. Because HARCOT charges wire delay against total declared area, compiling them out via `std::conditional_t` reduced floorplan area and recovered 18 ps on P2 (2.000 $\rightarrow$ 1.940 cycles), which enabled GHIST=160. This is a property of the cost model, not a predictor mechanism, and is reported so that other